

Decentralized Network Coding

Christina Fragouli¹
School of Comput. and Commun. Sciences
EPFL, Lausanne
christina.fragouli@epfl.ch

Emina Soljanin²
Bell Labs, Lucent
Murray Hill, NJ
emina@lucent.com

Abstract — This paper proposes deterministic algorithms for decentralized network coding. Decentralized coding allows to locally specify the coding operations at network nodes without knowledge of the overall network topology, and to accommodate future changes in the network such as addition of receivers. To the best of our knowledge, these are the first deterministic decentralized algorithms proposed for network coding.

I. INTRODUCTION

The famous min-cut, max-flow theorem states that a source node can send a commodity through a network to a sink node at the rate determined by the flow of the min-cut separating the source and the sink. Recently it has been shown that by linear re-encoding at nodes in communications networks, the min-cut rate can be also achieved in multicasting to several sinks (see [1] for linear network coding and references therein for the papers which introduced the subject). Constructing such coding schemes efficiently is the subject of current research [2]–[5].

Decentralized codes refer to codes that can be defined without the knowledge of the overall network topology and can be easily extended to accommodate future changes in the network such as addition of receivers. Such coding schemes are particularly desirable in practice, but have not yet received adequate attention in the literature. The deterministic network code design methods proposed so far result in codes that may need to be completely redesigned to accommodate the addition of a single user. Randomized code constructions recently proposed in [4] alleviate this problem, at the cost of an error probability.

The main contributions of this paper are in the design of network codes suitable for applications that require decentralized or local coding strategies. We propose a deterministic method to design decentralized codes, which is based on the information flow decomposition [6, 7]. To the best of our knowledge these are the first such algorithms proposed for network coding. Section II reviews notation and the background from information flow decomposition we will need. Section III and Section V present our algorithms.

II. THE NETWORK CODING MODEL

We consider a communications network represented by a directed acyclic graph $G = (V, E)$ with unit capacity edges. There are h unit rate information sources S_1, \dots, S_h and N receivers R_1, \dots, R_N . The number of edges of the min-cut between the source and each receiver node is h .

¹C. Fragouli was supported by Swiss National Foundation Grant No. 200021-103836/1.

²E. Soljanin was in part supported by the NSF grant No. CCR-0325673.

We denote by (S_i, R_j) , $1 \leq i \leq h$, a set of h edge disjoint paths from the sources to the receiver node j . Our object of interest is the subgraph G' of G consisting of the hN paths (S_i, R_j) , $1 \leq i \leq h$, $1 \leq j \leq N$. A way to specify a network code is to describe which operations each node in G' has to perform on its inputs for each of its outgoing edges.

We assume that source S_i emits σ_i which is an element of some finite field \mathbb{F}_q . In linear network coding, through each edge of G' flows a linear combination of source symbols, *e.g.*, the symbol flowing through some edge e of G' is given by

$$\alpha_1(e)\sigma_1 + \dots + \alpha_h(e)\sigma_h = \underbrace{[\alpha_1(e) \dots \alpha_h(e)]}_{c(e)} \begin{bmatrix} \sigma_1 \\ \vdots \\ \sigma_h \end{bmatrix},$$

where vector $c(e) = [\alpha_1(e) \dots \alpha_h(e)]$ belongs to an h dimensional vector space over \mathbb{F}_q . We shall refer to the vector $c(e)$ as the *coding vector* of edge e . Note that the coding vector of an output edge of a node has to lie in the linear span of the coding vectors of the node's input edges. To describe a network code, it is sufficient to specify the coding vector for each edge of the network.

The coding vectors associated with input edges of a receiver node define the system of linear equations that the receiver needs to solve to determine the source symbols. We refer to an assignment of coding vectors such that each receiver has a full rank of equations to solve as a *valid* network code.

A. Subtree Decomposition

To illustrate this discussion, we will use as an example the network with two sources multicasting to the same set of three receivers shown in Fig. 1. Consider the line graph γ of $G' =$

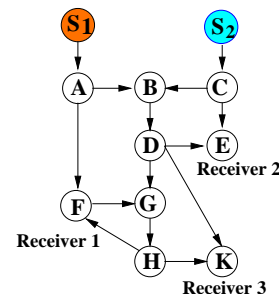


Figure 1: Network with two sources $\{S_1, S_2\}$ and three receivers $\{F, E, K\}$.

(V', E') , that is, the graph with vertex set E' in which two vertices are joined if and only if they are adjacent as edges in G' . The graph γ for our example network of Fig. 1 is depicted in Fig. 2a. Without loss of generality we can assume that the line graph contains a node corresponding to each of the h sources. We refer to these nodes as *source nodes*. Each node

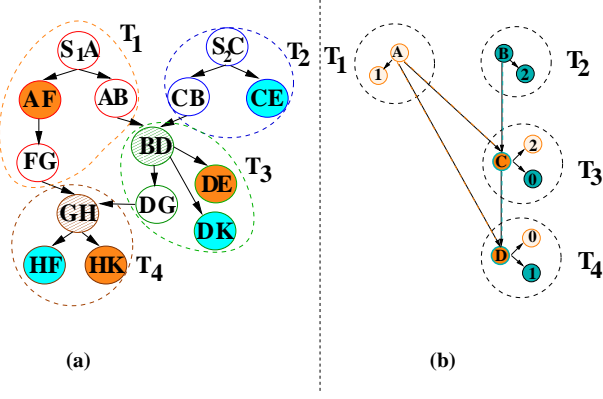


Figure 2: Line graph for the network with 2 sources and 3 receivers with coding points BD and GH and its subtree graph.

with a single input edge merely forwards its input symbol to its output edges. Each node with two or more input edges performs a coding operation (linear combining) on its input symbols and forwards the result to its output edges. We refer to these nodes as *coding points*:

Definition 1 Coding points are the nodes of γ with two or more inputs.

We refer to the node corresponding to the last edge of the path (S_i, R_j) as the *receiver node* for receiver R_j and source S_i . For a configuration with h sources and N receivers, there exist hN receiver nodes. For example, in Fig. 2a, S_1A and S_2C are source nodes, BD and GH are coding points, and AF, HF, HK, DK, DE and CE are receiver nodes.

We partition the line graph into subsets T_i so that each T_i contains exactly one source node or a coding point, and every other node belongs to the T_i containing its first ancestral coding point or source node. We shall call the subset T_i a *source subtree* if it starts with a source node or a *coding subtree* if it starts with a coding point.

The following properties of the subtree graph Γ follow directly from Definition 1 and the fact that the min-cut condition is satisfied in the original network G :

Theorem 1 The subtree graph satisfies the following:

1. Each T_i is a tree because the only nodes with two or more input edges in the line graph are the coding points.
2. For each receiver R_j , the h paths from the h source nodes to the h receiver R_j nodes are both edge and vertex disjoint.
3. For each receiver R_j , the h receiver nodes corresponding to the last edges on the paths (S_i, R_j) , $1 \leq i \leq h$, belong to distinct subtrees.
4. Each subtree contains at most N receiver nodes.

For the network code design problem, we only need to know how the subtrees are connected and which receiver nodes are in each T_i , whereas the structure of the network inside a subtree does not play any role. Thus we can contract each subtree to a node and retain only the edges that connect the subtrees, to get the *subtree graph* Γ . Fig. 2b shows the subtree graph for the network in Fig. 1; there are four subtrees:

T_1 and T_2 are source subtrees, T_3 and T_4 are coding subtrees. Network coding assigns an h -dimensional coding vector $c(T_i) = [c_1(T_i) \dots c_h(T_i)]$ to each subtree T_i .

B. Minimal Subtree Graphs and Their Properties

In the subtree graph Γ , for each receiver R_j , the h paths from the h source nodes to the h receiver R_j nodes are both edge and vertex disjoint. We will call this property the *multicast property* of the subtree graph.

Definition 2 A subtree graph is called minimal with the multicast property if removing any edge would violate the multicast property.

For example, the subtree graph in Fig. 2b is minimal. The following Theorem 2 follow directly from Definition 2, while Theorem 3 describes additional structural properties of minimal subtree graphs with two sources.

Theorem 2 For a minimal subtree graph, the following holds:

1. There does not exist a valid network code where the vectors assigned to the parents of any given subtree are not linearly independent.
2. There does not exist a valid network code where the vector assigned to a child belongs to a proper subspace of the space spanned by vectors assigned to its parents.
3. Each coding subtree has at most h parents.

Theorem 3 For a minimal subtree graph of a network with 2 sources and N receivers, the following holds:

1. A parent and a child subtree have either a child or a receiver in common.
2. Each coding subtree contains at least two receiver nodes.
3. Each source subtree contains at least one receiver node.
4. There exist at most $N - 1$ coding subtrees.

Proof of Theorem 3:

1. If a parent and a child subtrees have neither a child nor a receiver in common, they can be assigned the same coding vector. This scenario contradicts Theorem 2.
2. If a coding subtree is a terminal node of the subtree graph, it has two receivers and no children. Consider a coding subtree T that is an inner node of the subtree graph. Let P_1 and P_2 be its parents. By claim 1), a parent and a child have either a receiver or a child in common. If T has a receiver in common with each of its parents, then T has two receivers, since each receiver is shared by exactly two subtrees. If T and say P_1 do not have receiver in common, then they have a child in common, say C_1 . Now, if T and C_1 do not have receiver in common, then they have a child in common. And so forth, following the posterity brunch of P_1 , one eventually reaches a child of T that is a terminal node of the subtree graph, and thus have no children. Consequently, T has to have a receiver in common with this subtree. Similarly, if T and P_2 do not have child in common, there will be a descendent of P_2 and child of T which must have a receiver in common with T .

3. If network coding is not required, each source subtree contains N receivers. If network coding is required, each source subtree will have at least one child. The proof that a source subtree has at least one receiver is based on the same reasoning as above for coding subtrees, except that there is only one brunch of posterity.
4. Recall that there are exactly $2N$ receiver nodes. The claim then follows directly from claims 2) and 3).

III. CODES FOR TWO SOURCES AND N RECEIVERS

Consider a communications network represented by a directed acyclic graph $G = (V, E)$ with unit capacity edges, $h = 2$ unit rate information sources S_1, S_2 and N receivers R_1, \dots, R_N .

To label the nodes of a subtree graph of a network with two sources, we can use the points on the projective line $\mathbb{PG}(1, q)$:

$$[0\ 1], [1\ 0], \text{ and } [1\ \alpha^i] \text{ for } 0 \leq i \leq q-2. \quad (1)$$

For a valid network code, it is sufficient and necessary that the coding vector associated with a subtree lie in the linear span of the coding vectors associated with its parent subtrees, and the coding vectors of any two subtrees having a receiver in common be linearly independent. Since any two different points on the line are linearly independent and each point on the line is in the span of any two different points on the line, both coding conditions are satisfied if each node in the subtree graph is assigned a unique point of the projective line $\mathbb{PG}(1, q)$. This is the basic idea of the following algorithm:

Algorithm 1

1. *Find Subtree Graph*: Identify in G a set of h edge disjoint paths (S_i, R_j) , $1 \leq i \leq h$, from the source to the receiver j , $j = 1 \dots N$. Let G' denote the subgraph of G consisting of the hN paths (S_i, R_j) , $1 \leq i \leq h$, $1 \leq j \leq N$. Denote by C the set of edges in G' that are coding points (see Definition 1), that is, edges where paths meet for the first time.
2. If $|C| \leq N-1$ go to step 3, else go to step 4.
3. *Find Minimal Subtree Graph*: Associate a weight $w(e)$ with each edge e in G' , $w(e) = c > 0$ if e is an incoming edge to any of the edges in C and zero otherwise. Let G'_w denote the resulting weighted graph. Identify in G'_w minimum-weight max-flow paths for each receiver.
4. *Label Subtrees*: Create a number of tokens, each token associated with a different point in $\mathbb{PG}(1, q)$, and forward the tokens from the sources towards the destinations. Each coding point (corresponding terminal in the network) gets hold a token and uses the respective coding vector as long as required, then releases it back in the network for possible reuse.

To summarize, this algorithm assigns to each different subtree a different vector in the set $\mathbb{PG}(1, q)$. Each receiver is going to observe two distinct such vectors, and have a full rank system of equations to solve to retrieve the source information.

Since a network with N receivers has at most $N+1$ subtrees (Theorem 3), we are going to need at most $N+1$ coding vectors. The projective line $\mathbb{PG}(1, q)$ supplies $q+1$ points, thus the field with $q = N$ elements has a sufficient size.

In the first step of the algorithm we identify the paths to use and the edges where we need to perform linear combinations. If the number of such edges happens to be smaller than N , we can directly proceed to labeling. If not, or if we wish to optimize for employed resources, we proceed to reduce the number of coding points. From definition 2, no edge in a minimal subtree graph can be removed without violating the multicast condition for at least one receiver. Thus the minimal subtree graph has only the required number of such edges. Associating a weight with the corresponding edges and performing a weighted max-flow algorithm over G' allows to use a smaller number of such edges.

In the last step of the algorithm, we propose a method to ensure that a distinct coding vector is mapped to each subtree. An alternative simple way to organize this mapping is described below.

Recall that for each subtree, we locally know which receivers it contains and which sources are associated with each receiver (at the terminal before the coding point). In networks with two sources, each subtree contains at least one receiver node associated with S_1 and at least one receiver node associated with S_2 . Let $\{R_{i_1}, R_{i_2}, \dots, R_{i_u}\}$, where $i_1 < i_2 < \dots < i_u$ be the set of receivers associated with S_1 in a given subtree. We choose $[1\ \alpha^{i_1-1}]$ to be the label of that subtree. This way no other subtree can be assigned the same label since the receiver R_{i_1} can be associated with the source S_1 in at most one subtree. Note that this is not the most efficient mapping as it may require alphabet of size q , with $q-2 = N-1$, as opposed to $q-1 = N-1$.

Algorithm 1 is suboptimal, in the sense that it employs a larger alphabet size than required. Indeed, in [7] it was shown that an alphabet of size $\Theta(\sqrt{N})$ is always sufficient for a network with two sources. However, it was also demonstrated in [8] that the problem of identifying the minimum alphabet size required for a given configuration is NP-complete.

Example 1 A decentralized code for the network in Fig. 1. Since we have three receivers, there will be at most two coding trees. We can use the following four points to label the subtrees T_1, T_2, T_3 , and T_4 in Fig. 2b:

$$[1\ 0], [0\ 1], [1\ 1], [1\ \alpha].$$

For the decentralized code, we need alphabet of size three. Note that there exists a valid network code with alphabet size two which assigns $[0\ 1]$ to T_4 .

Comparing with algorithms in the literature, we note first that the described algorithm applies in the special case of $h = 2$ sources (we will discuss extensions to $h > 2$ in the following section). Given this observation, randomized algorithms entail a probability of error and employ a much larger alphabet [4]. The polynomial time algorithms proposed in [2, 3] also use alphabet size of N , however the algorithm involves, instead of steps 3 – 5, to visit every edge in the graph and calculate an eligible vector with respect to a candidate basis carried along for each receiver, which may become impractical as the number of edges and receivers grows. Moreover, the algorithms in [2, 3] are not decentralized.

One of the main advantages of decentralized codes is that they do not have to be changed with the growth of the network as long as the subtree decomposition remains the same, or the new subtree graph contains the original subtree graph.

To achieve that, since the coding vectors associated with any two subtrees provide a basis of the 2-dimensional space, we can think of subtrees as “secondary sources” and allow new receivers to connect to any two different subtrees. Thus we can extend our network, without perturbing the already existing users.

Note that, the projective line $\mathbb{PG}(1, q)$ forms a subset of the projective line $\mathbb{PG}(1, q')$ if $\mathbb{F}_{q'}$ is an extension field of \mathbb{F}_q . Thus, if we need to create additional coding vectors to allocate to new subtrees, we can employ unused points from the projective line $\mathbb{PG}(1, q')$.

IV. CODES FOR h SOURCES AND 2 RECEIVERS

From the alphabet bounds derived in [2] and [3] on codes using global information, we know that there are valid binary codes for networks with h sources and 2 receivers. We here show that there is only one valid binary code assignment for the minimal subtree graph of a network with 2 receivers, and that this assignment needs only local information.

Since $N = 2$ each coding subtree has at most two receiver nodes, and thus, because of the multicast (min-cut) condition, each coding subtree has exactly two inputs.

Theorem 4 *The binary code that assigns to each source subtree, a different basis vector, and to a coding subtree, the binary sum of the vectors assigned to its two parents is the only valid binary code for the minimal subtree graph of a network with 2 receivers.*

Proof of Theorem 4:

From Claim 1. of Theorem 2, we know that there does not exist a valid network code where a subtree is assigned the same coding vector as one of its parents. Therefore, since the code is binary and there are exactly two parents, a code subtree must be assigned the binary sum of the vectors assigned to its two parents. Therefore, this is the only code that satisfies a necessary condition for validity. Since there exist binary codes for networks with 2 receivers, the code must be valid.

V. CODES FOR h SOURCES AND N RECEIVERS

The common theme in the following algorithms is that, to simplify the code design problem, apart from requiring that the mincut to each receiver is h , we can also require some additional structure. For example, if we require that the mincut to each node of the graph is h , then we can find h edge-disjoint trees that span the source and the destination nodes, and thus no network coding is required. The following algorithms impose less severe requirements.

Algorithm 2

A straightforward algorithm in the case where h is an even number, is to decompose the problem in $h/2$ two-source configurations and then apply Algorithm 1 to each one separately. For such a decomposition to exist, the min-cut from every pair of sources to each receiver has to be two, and paths corresponding to sources outside the pairs cannot overlap. Thus, this is obviously a suboptimal algorithm that will require increased resources (additional edges). However, even this very simple suboptimal algorithm can offer significantly throughput benefits as compared to not using network coding. Following we illustrate this point.

Theorem 5 *Consider a network with two sources and N receivers where the multicast condition is satisfied for each receiver. Let t_i denote the throughput for receiver i and $t = \sum_{i=1}^N t_i$ the total aggregate throughput. If network coding is employed, $t = 2N$, while if network coding is not employed, $N + 1 \leq t \leq 2N$.*

Proof of Theorem 5:

Since the mincut to each node is one, there exists one tree that spans the source and the destination nodes. Moreover, since each source subtree contains at least one receiver node (Theorem 3), at least one of the receivers will be able to receive both sources. Thus a lower bound on the achievable throughput is $N + 1$. For every N , there do exist minimal configurations where without network coding we can not achieve throughput better than $N + 1$. These are all the subtree graphs with $N - 1$ coding subtrees, where each coding subtree has a different pair of parents. The subtree graph in Fig. 2b falls in this category.

Corollary 1 *There exist configurations with h sources where if network coding is not employed, the total aggregate throughput is $\frac{h}{2}(N + 1)$, while employing Algorithm 2 allows to achieve throughput hN .*

Algorithm 3

The basic idea in this algorithm is to use as coding vectors points of arcs.

Definition 3 *In $\mathbb{PG}(h - 1, q)$, a k -arc is a set of k points any h of which form a basis for \mathbb{F}_q^h .*

For introduction to arcs, see [9] and references therein. In combinatorics, arcs correspond to sets of vectors in general position.

Definition 4 *The vectors over \mathbb{F}_q^h in a set \mathcal{A} are said to be general position if any h vectors in \mathcal{A} are linearly independent.*

Example 2 *The following set of $h + 1$ points are in general position in \mathbb{F}_q^h , and form an arc in $\mathbb{PG}(h - 1, q)$ known as the normal rational curve:*

$$\begin{pmatrix} 1 & x_1 & x_1^2 & \dots & x_1^{h-1} \\ 1 & x_2 & x_2^2 & \dots & x_2^{h-1} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & x_q & x_q^2 & \dots & x_q^{h-1} \\ 0 & 0 & 0 & \dots & 1 \end{pmatrix}, \quad (2)$$

where $x_i \in \mathbb{F}_q$ and $x_i \neq x_j$ for $i \neq j$. Note that by an appropriate change of basis, this set of points can be mapped into an arc containing h vectors that form the identity matrix I_h and $q + 1 - h$ vectors with no component equal to 0.

Arcs are of special interest for us because they enable decentralized network coding. Namely, as long as we take a point from an arc to be a coding vector of a subtree, we do not need to know other subtrees' coding vectors (as long as they are different points of the same arc) or the structure of the network. We do, however, have to know the coding vectors of the parent subtrees, to associate with the child subtree a coding vector in their span.

One way to overcome this problem, is to artificially create subtree graphs where each subtree has exactly h parents, so

that the coding vectors of the parent subtrees form a basis of the h -dimensional space. That is, we require that the mincut towards each coding point is h , and have at the input of each subtree incoming coding vectors that form a basis of the h -dimensional space. Similarly to Theorem 3 we can show the following:

Theorem 6 *In a minimal subtree graph of a network with h sources and N receivers, if all coding subtrees have exactly h parents, then each subtree contains at least h receiver nodes, and thus the total number of coding subtrees is at most $N - 1$.*

Corollary 2 *In a minimal subtree decomposition of a network with h sources and N receivers, if all coding subtrees have exactly h parents, the alphabet of size $N + h - 2$ is sufficiently large for decentralized coding.*

Indeed, we can use the $N + h - 1$ points in the normal rational curve (see Example 2) in $\mathbb{PG}(h - 1, N + h - 2)$ to assign h vectors to the source subtrees and $N - 1$ vectors to the coding subtrees.

There are many possible variations in implementing this idea. For example, we can first identify a minimal subtree configuration, and then join the coding points that have less than h inputs to the corresponding sources. Or, instead of identifying a minimal configuration, we can simply use an alphabet large enough for the number of coding points in our network.

Algorithm 4

An alternative method to handle constraints on the set of coding vectors \mathcal{A} is to develop algorithms to generate appropriate \mathcal{A} 's or use theorems about geometry of arcs to derive \mathcal{A} 's.

Both approaches are illustrated below on the special case of networks with three sources and N receivers where no coding subtree has a child. Thus the subtree graph is bipartite with one set of nodes consisting of the three source subtrees and another set of nodes consisting of coding subtrees.

To the source subtrees, we assign the basis vectors $e_1 = [1\ 0\ 0]$, $e_2 = [0\ 1\ 0]$ and $e_3 = [0\ 0\ 1]$. Depending on which source subtrees a coding subtree has as its parents, we assign to it a vector belonging to one of the following sets:

$$\mathbb{P}(\langle e_1, e_2 \rangle), \mathbb{P}(\langle e_1, e_3 \rangle), \mathbb{P}(\langle e_2, e_3 \rangle), \text{ and } \mathbb{P}(\langle e_1, e_2, e_3 \rangle)$$

where $\langle x, y \rangle$ denotes the span of vectors x and y , and $\mathbb{P}(X)$ denotes the projective space of the vector space X . Since each coding subtree contains at least two receivers, the maximum number of subtrees is $\lfloor 3N/2 \rfloor - 1$. The sets have to satisfy the following conditions:

1. Each set contains $N + 1$ elements.
2. Any three vectors in Π_{123} are linearly independent.
3. Any three different vectors such that not all of them belong to the same set are linearly independent.

A computer search provided several such sets for different numbers of receivers, and we also have the following explicit construction. We start with a finite field \mathbb{F}_q with a characteristic different than 2, and its quadratic extension \mathbb{F}_{q^2} . For set

Π_{123} , we take elements of the form $[1\ \sqrt{\alpha^i}\ \alpha^i]$ such that $\alpha^i \in \mathbb{F}_q$ but $\sqrt{\alpha^i} \notin \mathbb{F}_q$; thus $\sqrt{\alpha^i} \in \mathbb{F}_{q^2}$ (see for example [10, Ch. 2]). For sets Π_{12} , Π_{23} , and Π_{13} , we take elements of the form $[1\ \alpha^i, 0]$, $[0\ 1\ \alpha^i]$ and $[1\ 0\ \alpha^i]$, respectively, such that $\alpha^i \in \mathbb{F}_q$. For the set Π_{13} , we use α^i such that $\sqrt{\alpha^i}\sqrt{\alpha^j} \neq \alpha^i$, for α^i, α^j employed in set Π_{123} . Additionally, we use a mapping such that not both the vectors $[1\ 0\ \alpha^k]$ and $[0\ 1\ \alpha^k]$ are used for the same configuration. Similarly for the vectors $[1\ 0\ \alpha^k]$ and $[1\ \sqrt{\alpha^k}\ \alpha^k]$. These requirements, however, lead to an increase of the required alphabet size.

Alternatively, given a set of constraints that coding vectors have to satisfy, we can also look for an appropriate arc tailored to the specific requirements as illustrated by the following example.

Example 3 *Suppose we need 6 3-dimensional vectors in general position such that: 2 are in Π_{12} , 2 in Π_{23} , 1 in Π_{13} , 1 in Π_{123} . Is there an arc of length 6 in $\mathbb{PG}(2, 4)$? In other words, we can we start with a known arc of length 6 in $\mathbb{PG}(2, 4)$ (such as the one on the left-hand side in the equation (3) below), and obtain the arc we are interested in by applying a projective transformation. In this particular case, the answer is positive, and the desired arc is obtained as follows:*

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & \alpha & \alpha^2 \\ 1 & \alpha^2 & \alpha \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & \alpha \\ 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & \alpha \\ 0 & 1 & 1 \\ 1 & 0 & \alpha^2 \\ 1 & 1 & 0 \\ 1 & 1 & \alpha^2 \end{bmatrix}. \quad (3)$$

REFERENCES

- [1] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inform. Theory*, Vol. 49, pp. 371–381, Feb. 2003.
- [2] P. Sanders, S. Egner, and L. Tolhuizen, "Polynomial time algorithms for network information flow," *Proc. 15th ACM Symposium on Parallel Algorithms and Architectures*, 2003.
- [3] S. Jaggi, P. Chou, and K. Jain, *Low complexity algebraic multicast network codes*, ISIT 2003, Japan 2003.
- [4] T. Ho, D. R. Karger, M. Médard, and R. Koetter, "Network Coding from a Network Flow Perspective", *IEEE International Symposium on Information Theory (ISIT)*, June 2003.
- [5] M. Feder, D. Ron, and A. Tavor, "Bounds on linear codes for network multicast", *Electronic Colloquium on Computational Complexity*, Report No. 33, 2003.
- [6] C. Fragouli and E. Soljanin "Information flow decomposition for linear network coding," preprint, submitted to *IEEE Trans. Inform. Theory*.
- [7] C. Fragouli, E. Soljanin and A. Shokrollahi, "Network coding as a coloring problem," CISS 2004.
- [8] A. Rasala-Lehman and E. Lehman, "Complexity Classification of Network Information Flow Problems", *SODA 2004*, pp. 142–150.
- [9] A. H. Ali, J. W. P. Hirschfeld, and H. Kaneta, "On the size of arcs in projective spaces," *IEEE Trans. Inform. Theory*, Vol. 41, pp. 1649–1656, Sept. 1995.
- [10] R. Lidl and H. Niederreiter, *Finite Fields*, New York: Cambridge Univ. Press, 1997.